

**REMARKS**

Claims 1-21 are pending in the application and stand rejected.

**Rejection under 35 U.S.C §102**

Claims 1-5, 8-17, 20 and 21 stand rejected under 35 U.S.C. 102(e) as being anticipated by the Wong reference. Applicants have reviewed the reference with care, paying particular attention to the passages cited, and are compelled to respectfully disagree with the Examiner's understanding of this reference. Because the Examiner appears to have overlooked the key differences between Wong and their invention, Applicants will first present a brief summary of the invention and then will identify and discuss each of these differences.

The invention recited in the claims is directed generally to logical key hierarchies ("LKH") such as 12<sub>0</sub> in Figure 2 – such a hierarchy is established for a group of clients, each of which is associated with a respective leaf key of the hierarchy. Each client knows its leaf key and all the other keys in the path from that leaf to the root of the hierarchy. The root key of the hierarchy is a group key and can be used to send a secure message to all the clients in the group. As described on page 1, as clients join and leave the group, it is necessary to update the keys and how this can be done is described in the IETF RFC 2627 referenced on page 1. Basically, updating involves the sending of update records each of which comprises a new key encrypted using a key that is a descendent of the new key in the LKH.

The invention deals with the problem of how to bring up-to-date a client that has been offline for a period and has therefore missed a number of update records. The solution is provided by the invention in the form of an entity (reference 20 in Figure 1) that receives all update records and consolidates them into a key history tree (the "key tree" recited in the claims) that can be used by a re-connecting client to catch up on the current group key. Applicants note that it is crucial to understanding the invention to appreciate the fact that this "key tree" (KHT) is not the same data structure as the logical key hierarchy (LKH). In the application, Figure 2

shows changes to the LKH as the result of the joining and leaving of various clients and depicts the various update records sent, and Figure 3 shows the corresponding changes to the KHT.

Thus, it is important to understand that the invention consolidates update records into a “key tree” as recited in claim 1:

“a manager for maintaining, on the basis of the received records, a key tree with nodes corresponding to nodes in said hierarchy,

the manager being arranged to store in association with each tree node, for each encrypting key used in respect of the encrypted key associated with the node, the most up-to-date version of the encrypted key and its version information with any earlier versions being discarded.”

Thus, the key tree (KHT) of the invention is built from the update records and one or more encryptions of the corresponding key of the key hierarchy (LKH) are stored in association with each node (one or more because there can be a respective encryption for each descendent node). In short, while the LKH is a hierarchy of unencrypted keys, one per node, the KHT is a hierarchy of encrypted keys, generally more than one per node.

The Wong reference, by its own admission (page 17, right col. ll. 6-12) covers the same ground as RFC2627 (mentioned above and on page 1 of the specification) - that is, logical key hierarchies (LKH) and their use in key management. Wong teaches certain enhancements to LKH (page 17, right col. ll. 13-37) but none of these relate to the problem addressed by the present invention, namely how to efficiently update a client who has been offline.

With specific reference to claim 1, Applicants note that Wong does not teach or suggest “a manager for maintaining, on the basis of the received records, a key tree with nodes corresponding to nodes in said hierarchy.” In Wong, the only entities present are a server  $s$  (corresponding to the LKH key manager 10 of Applicants’ Figure 1) and users  $u_i$  onwards (corresponding to the clients 14 of Applicants’ Figure 1). This server  $s$  maintains in memory the logical key hierarchy (called “group key graph” in Wong – page 19, left col. l. 25) and generates the key update messages (called “rekey messages” in Wong – page 20, right col. first paragraph)

for updating the users. The server  $s$  does not, however, maintain the group key graph on the basis of received rekey messages but, rather, generates the rekey messages on the basis of the group key graph. The server  $s$  therefore clearly does not read upon the “manager” recited in claim 1 because each user  $u$  is only interested in the keys lying in the path from an associated leaf node of the group key graph and the root of the graph. No user needs to construct a key tree with nodes corresponding to the group key graph and Wong does not disclose the construction of such a graph by a user.

Further with respect to claim 1, Wong also does not teach or suggest “the manager being arranged to store in association with each tree node, for each encrypting key used in respect of the encrypted key associated with the node, the most up-to-date version of the encrypted key and its version information with any earlier versions being discarded.” The rekey messages of Wong, like the claimed key updates, comprise keys encrypted by other keys (see, e.g., page 20, right col. first paragraph). However, there is no disclosure in Wong of maintaining a tree and storing in association with the nodes of this tree, the encrypted keys received in rekey messages.

For all of the above reasons, Applicants respectfully submit that claim 1 is in fact novel over Wong and request the Examiner to kindly reconsider and pass this claim to issue. Applicants further submit that the above discussion is equally probative of the novelty of independent claims 13 and 21, and thus respectfully request that claims 13 and 21 be passed to issue as well.

Claims 2-5 and 8-12 depend from claim 1, and claims 14-17 and 20 depend from claim 13. Applicants thus submit that claims 2-5, 8-12, 14-17 and 20 are also allowable over the art on record at least by virtue of their dependencies as well as the additional limitations recited by each of these claims.

#### Rejection under 35 U.S.C §103

Claims 6, 7, 18 and 19 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Wong in view of U.S. Pat. No. 6,606,706 to Li. Claims 6 and 7 depend from claim 1 and claims 18 and 19 depend from claim 13. “If an independent claim is nonobvious under 35 U.S.C. 103,

then any claim depending therefrom is nonobvious.” *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988). Therefore, in light of the above discussion of claim 1, Applicants submit that claims 6, 7, 18 and 19 are also allowable at least by virtue of their dependencies as well as the additional limitations recited by each of these claims.

~ ~ ~

Regarding the prior art made of record by the Examiner but not relied upon, Applicants believe that this art does not render the pending claims unpatentable.

In view of the above, Applicants submit that the application is now in condition for allowance and respectfully urge the Examiner to pass this case to issue.

\* \* \*

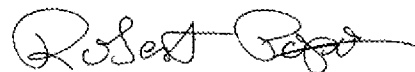
The Commissioner is authorized to charge any additional fees which may be required or credit overpayment to deposit account no. 08-2025. In particular, if this response is not timely filed, the Commissioner is authorized to treat this response as including a petition to extend the time period pursuant to 37 CFR 1.136(a) requesting an extension of time of the number of months necessary to make this response timely filed and the petition fee due in connection therewith may be charged to deposit account no. 08-2025.

I hereby certify that this document is being transmitted to the  
Patent and Trademark Office via electronic filing.

July 31, 2008

(Date of Transmission)

Respectfully submitted,



Robert Popa  
Attorney for Applicants  
Reg. No. 43,010  
(323) 934-2300 voice  
(323) 934-0202 facsimile  
rpopa@la.ladas.com